Saturated random packing built of arbitrary polygons under random sequential adsorption protocol

Michał Cieśla [*] and Piotr Kubala[†]

M. Smoluchowski Institute of Physics, Department of Statistical Physics, Jagiellonian University, Łojasiewicza 11, 30-348 Kraków, Poland

Ge Zhang[‡]

Department of Physics, University of Pennsylvania, Philadelphia, Pennsylvania 19104, USA

Random packings and their properties are a popular and active field of research. Numerical algorithms that can efficiently generate them are useful tools in their study. This paper focuses on random packings produced according to the random sequential adsorption (RSA) protocol. Developing the idea presented by G. Zhang [Phys. Rev. E **97**, 043311 (2018)], where saturated random packings built of regular polygons were studied, we create an algorithm that generates strictly saturated packings built of any polygons. Then, the algorithm was used to determine the packing fractions for arbitrary triangles. The highest mean packing density, $0.552\,814 \pm 0.000\,063$, was observed for triangles of side lengths $0.63 : 1 : 1$. Additionally, microstructural properties of such packings, kinetics of their growth, as well as distributions of saturated packing fractions and the number of RSA iterations needed to reach saturation were analyzed.

I. INTRODUCTION

Random sequential adsorption (RSA) is a numerical protocol to generate random packings of particles with a given shape [1,2]. It is based on sequential trials to add a randomly placed particle to a packing. The particle is added when it does not intersect with any previously placed particle; otherwise, it is removed. When added, it will hold its position and orientation inside the packing forever. The process is continued until the packing is saturated, i.e., there is no area large enough to place any new particle.

The history of RSA began in 1939 when Flory was studying random attachment of atomic groups to a long polymer. In 1959, Renyi solved the so-called "car parking problem" [3], which was a one-dimensional implementation of RSA. The popularity of RSA, however, is owing to Feder, who noted that RSA packings resemble the structure of monolayers created during irreversible adsorption processes [4]. RSA is also studied as one of the simplest yet not trivial protocols to generate random packings, which takes into account excluded volume effects. In contrast to random close packings, which are the most popular models of granular matter, saturated RSA packings have well-defined mean densities [5].

One of the biggest issues of the RSA protocol is its inefficiency for almost saturated packings. When the probability of successfully adding another particle to the packing is tiny, the algorithm needs a tremendous amount of random trials to insert any new particle. Moreover, even if the packing is already saturated, it is often difficult to ascertain saturation in order to stop trying to add subsequent particles to the packing. The most common way to deal with this problem is to stop the simulation after some large but arbitrarily defined number of iterations and then estimate interesting properties of saturated packings (e.g., its density) by extrapolating packing growth kinetics [6,7]. Although such an approach can give pretty accurate results [8], it requires a large amount of computer resources and still is burdened with systematic error due to the estimation. The best method so far to deal with this problem is to trace regions where the next particle can be added. It allows to significantly speed up packing generation because subsequent attempts to place an object can be restricted only to these regions. Moreover, when all such regions vanish, insertion attempts may cease since the packing is known to be saturated. The method is illustrated in Fig. 1 for two-dimensional disks.

The method of tracing available regions has been successfully applied for packing built of spherically symmetric shapes [9–11] and, recently, for ellipses and spherocylinders [12], regular polygons [13], rectangles [14], and any shapes built of several disks [15].

The aim of this study is to show how the idea used for generating saturated random packing built of regular polygons [13] can be expanded to work for any polygon. It is worth noting, that together with a recent algorithm presented in Ref. [15], this method allows one to generate strictly saturated RSA packings for a large range of two-dimensional shapes. Therefore it enables us to effectively model the majority of monolayers produced in irreversible adsorption experiments. For example, the presented algorithm is used here to determine the shape of triangles that build the densest RSA packing. Although the RSA packings built of triangles do not have a direct application in modeling adsorption monolayers of realistic molecules, their study may help answer some

---
[*]michal.ciesla@uj.edu.pl
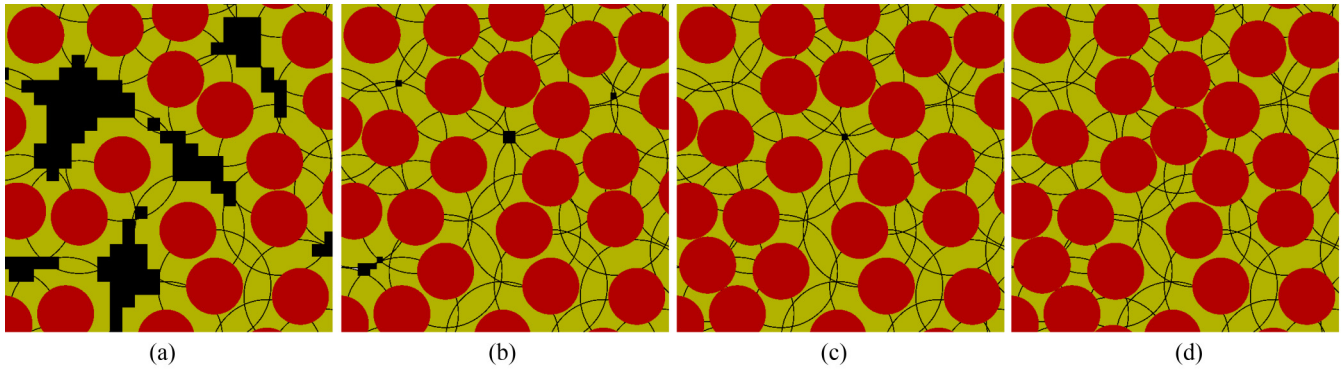[†]pkua.log@gmail.com
[‡]gezhang@alumni.princeton.edu

FIG. 1. Stages involved in generating an RSA packing of 2D disks. Red disks represent nonpenetrable particles. Yellow indicates regions where it is impossible to place the center of another disk due to intersection with the existing ones. Black square voxels approximate an area where placing of a subsequent disk is possible. In each subsequent stage, these voxels are divided into 2^d subvoxels to better approximate available space as well as are filled up with new disks. In panel (d), there are no black voxels; thus, the packing is saturated.

fundamental questions regarding random packing properties. Firstly, it was observed that shapes of higher symmetry, such as disks or spheres, form less dense packings than anisotropic shapes such as ellipses or ellipsoids [12,16–20]. Triangles do not have continuous symmetry groups such as disks or spheres, but they may have D_3 dihedral symmetry (equilateral triangles) or axial symmetry (isosceles triangles). It is important to check if these symmetries influence the properties of the obtained packings. It was observed that in the case of rectangles, the local minimum of the packing fraction is observed for squares [14,16]. Another recent study of random packings built of shapes with tetrahedral and octahedral symmetry shows that it is not true that octahedral symmetry will always lead to less dense packings than tetrahedral symmetry [21]. Secondly, recent analytical results obtained by Baule shows that the kinetics of packing growth near saturation limit may depend on properties of the contact function—the touching distance between two objects as a function of their relative orientation [22]. For example, the long-time scaling of the kinetics is different between packings built of ellipses and rectangles with centers placed on a one-dimensional line. This has not been observed so far for two-dimensional packings that are almost saturated [8,17,23] but now, using the algorithm that generates strictly saturated packings, it is possible to check this arbitrarily close to saturation limit. Additionally it would be possible to check if the scaling of the number of RSA iterations needed to reach saturation is governed by the same exponent as the kinetics of packing growth [24].

II. ALGORITHM

The crucial stage of the algorithm is when it determines if a given voxel can be eliminated due to the impossibility to add another particle inside it. In the case of disks or spherically symmetric objects, a voxel can be eliminated if the distance from each of its vertices to the center of any already added disk is smaller than $2r$, where r is the disk's radius [9,11,12,14]. Because a voxel is convex, it is removed when all its corners are inside the excluded area around an already placed particle. Another analytic approach is to define

the intersection function and estimate its behavior inside a voxel. For example, in the case of the disk, such function can be defined as

$$F_{\text{disk}}(x, y) = (x - x_0)^2 + (y - y_0)^2 - 4r^2, \quad (1)$$

where (x_0, y_0) are coordinates of the disk center from the packing, and r is its radius. Thus, the voxel v can be removed only if for all $(x, y) \in v$, $F_{\text{disk}}(x, y) < 0$, or equivalently

$$\max_{(x,y)\in v} F_{\text{disk}}(x, y) < 0. \quad (2)$$

Although, in general, finding the maximum of $F(x, y)$ for (x, y) inside a voxel analytically can be difficult, it is enough to use an upper limit of $F(x, y)$: $\tilde{F}(x, y) \geqslant F(x, y)$, as long as $\tilde{F}(x, y)$ tends to $F(x, y)$ when the voxel size goes to 0. The Zhang algorithm used for generating saturated RSA packings of two-dimensional regular polygons [13] employs this approach. The following is its brief summary. Two polygons intersect only if any of their edges intersect. Assume that there are two line segments, A and B, each defined by two vectors pointing at their two ends, $(\mathbf{a_1}, \mathbf{a_2})$ and $(\mathbf{b_1}, \mathbf{b_2})$, respectively. Segments A and B intersect if and only if the ends of segment A lay on the opposite sides of the line containing segment B, and vice versa. Thus,

$$F_1(\mathbf{a_1}, \mathbf{a_2}, \mathbf{b_1}, \mathbf{b_2}) = f(\mathbf{a_1}, \mathbf{a_2}, \mathbf{b_1}) \cdot f(\mathbf{a_1}, \mathbf{a_2}, \mathbf{b_2}) < 0 \quad \text{and}$$

$$F_2(\mathbf{a_1}, \mathbf{a_2}, \mathbf{b_1}, \mathbf{b_2}) = f(\mathbf{b_1}, \mathbf{b_2}, \mathbf{a_1}) \cdot f(\mathbf{b_1}, \mathbf{b_2}, \mathbf{a_2}) < 0, \quad (3)$$

where $f(\mathbf{a}, \mathbf{b}, \mathbf{c}) = (\mathbf{c} - \mathbf{a}) \cdot (\mathbf{b} - \mathbf{a})$ is positive if the point $\mathbf{c}$ is on one side of the line passing through $\mathbf{a}$ and $\mathbf{b}$, and negative if $\mathbf{c}$ is on the other side.

When generating two-dimensional RSA packings of polygons, the placement of an incoming particle can be specified by three scalars: the coordinate of its center, (x, y), and the angle between the orientation of the polygon and a reference orientation, ϕ. Thus, each point of a three-dimensional auxiliary space (x, y, ϕ) corresponds to an insertion of a trial polygon in this particular location and orientation. Three-dimensional voxels are employed to trace parts of this space where successful insertion trials are possible. In subsequent stages of the algorithm, some of these voxels are covered by newly added polygons; others are divided into 2^3 subvoxels to
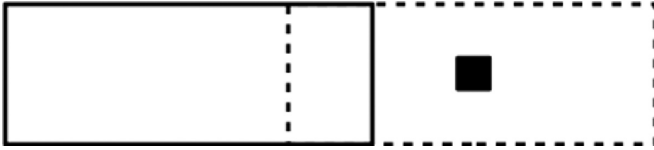
FIG. 2. Example of a voxel (the black square) that would never be removed by the algorithm described in [13]. The solid rectangle is already part of the packing while the dashed one is the virtual one with the center inside the black square voxel. Rectangles partially overlap but their edges do not, according to the criterion (3).
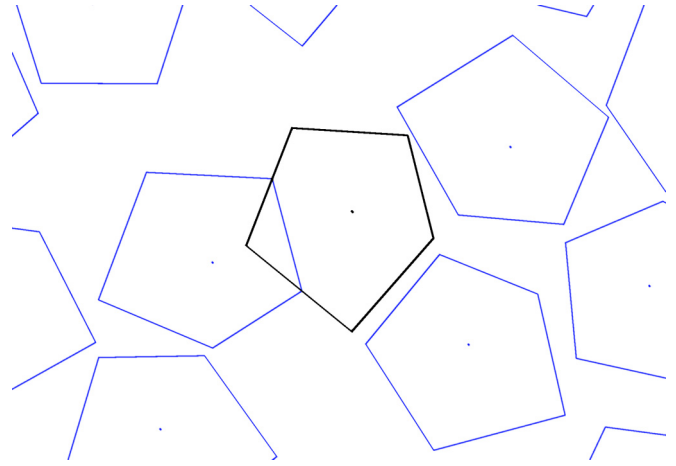

FIG. 3. An unavailable voxel that could not be identified as such using the original algorithm [13]. Blue pentagons are objects already inserted into the packing and the black pentagon represents the unavailable voxel.

approximate the available space better. As mentioned above, the crucial stage of the algorithm is determining if a voxel can be removed, as inserting a new polygon inside any of its points leads to an intersection with previously placed objects. The algorithm identifies polygon overlaps by checking if any sides of two polygons overlap using Eq. (3). To determine if a given voxel overlaps with an existing polygon, the algorithm treats the voxel as a trial insertion with an uncertainty ($x \pm \delta x, y \pm \delta y, \phi \pm \delta \phi$), and performs a worst-case error analysis of the intersection criterion. The analysis is based on the functions $\delta F_i(\mathbf{a_1}, \mathbf{a_2}, \mathbf{b_1}, \mathbf{b_2})$, which provide the error bounds of conditions (3) for edge A belonging to a polygon already added to the packing and edge B from a trial particle within the voxel [see Eqs. (8)–(15) in Ref. [13]]. When $\tilde{F}_i = F_i + |\delta F_i| < 0$, an intersection always exists no matter how $\delta x, \delta y$, and $\delta \phi$ vary within their respective limits (determined by the voxel size); and thus the entire voxel can be removed. Additionally, if condition (1) is fulfilled for the existing polygon's inscribed circle, a voxel is removed too.

It turns out that for nonregular polygons, this algorithm requires a modification. If one employs the original algorithm, certain unavailable voxels can never be correctly identified. For example, the black voxel in Fig. 2 will never be removed as it contains a point (corresponding to the center of the dashed rectangle) for which the functions $F_i = 0$. Therefore, even if $|\delta F_i| \to 0$ with decreasing voxel size, the maximum over the voxel is $\tilde{F}_i \geqslant 0$. Thus, the voxel will not be eliminated even though there is no possibility to place the nonintersecting rectangle in it. A similar situation is shown in Fig. 3, where the edges of one object cross the vertices of the other one. While both objects intersect, their edges, according to (3), do not. Again, a small voxel containing the center of the black polygon should be removed because the black polygon intersects with the existing blue polygon. However, the program could not ascertain its unavailability since the black polygon's sides only touch the sides of the blue polygon. In this case, the functions F_i are again equal to zero, and their upper bound $F_i + \delta F_i \geqslant 0$. Thus, this voxel will not be eliminated and the algorithm will never stop. It is theoretically possible that such trial object will be randomly selected, which would lead to packing with overlapped objects. Practically, the probability of such selection is negligible $\sim (1/2^{53})^3$, assuming that three double-precision variables determine the position and orientation of the object. Even if strong inequalities in (3) are replaced by weak ones, which is a protection against overlapping of packed objects, the problem with voxels that can never be removed still remains.

In this paper, we present a way to fully overcome this issue. We place so-called "helper segments," which are additional line segments, within a polygon. Although there might be many different ways of adding helper segments to overcome this problem, here we present two generic and numerically efficient methods. The first method is to add helper segments from an arbitrary point inside the polygon to all vertices. The second method is to add a parallel helper segment behind each of the polygon's edges. The distance between the edge and the helper segment parallel to it should be smaller than the distance between any two vertices. Our experience shows that the first method is generally faster, since the helper segments share one endpoint with polygon edges, and therefore do not need to be recalculated. However, the second method may be required when the first method is inapplicable. (For example, if it is impossible to find a point inside a particular nonconvex polygon such that all helper segments from the point to its vertices are inside the polygon). The two ways to add helper segments are illustrated in Fig. 4.

Another possibility is to treat all polygon diagonals as helper segments, which works well for rectangles. However, the number of diagonals grows quadratically with the number of vertices, which significantly reduces the efficiency of the algorithm for polygons with a high number of vertices.
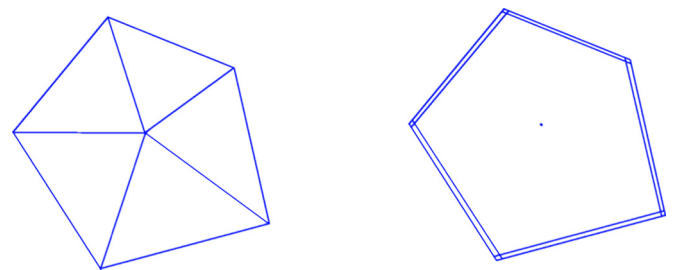

FIG. 4. Illustration of the two ways to add helper segments to an irregular pentagon to solve the problem detailed in the text.
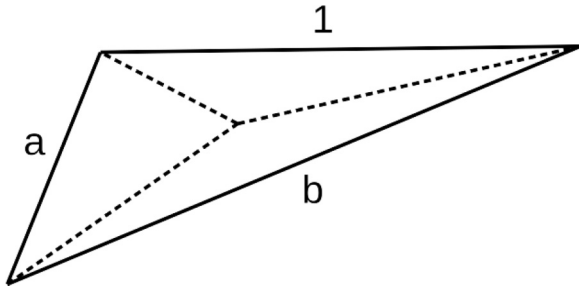
FIG. 5. Model of a triangle of side lengths ratio $a \leqslant 1 \leqslant b$. Dashed lines are helper segments from the triangle mass center to its vertices.
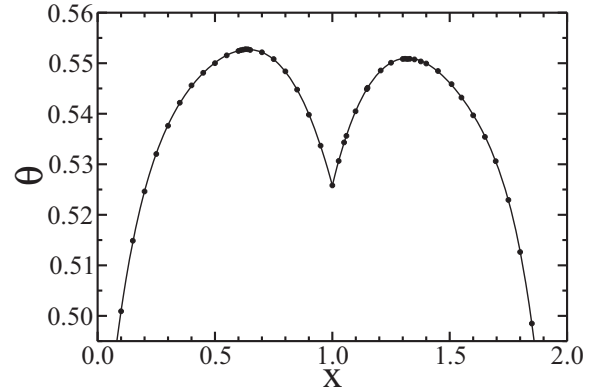

FIG. 7. The mean saturated packing fraction for isosceles triangles. The parameter x denotes the length ratio of the isosceles triangle base to its arm, e.g., $x = 1$ corresponds to equilateral triangle. Dots are numerical results, and the solid line is to guide the eye. Statistical errors are smaller than the size of dots.

III. TEST CASE—RSA OF TRIANGLES

To prove the correctness of the presented algorithm, saturated packings built of identical triangles were generated in order to find a shape that provides the densest packings.

A. Model

The shape of an arbitrary triangle is fully determined by three numbers corresponding to its side sizes. Without any loss of generality, the length of the middle side can be taken as the unit of length. Thus, there are only two independent variables a and b such that $a \leqslant 1 \leqslant b$, and additionally, due to triangle inequality condition, $a + 1 \geqslant b$ (see Fig. 5). Triangles were rescaled to have a unit area, and were put randomly on a square box of size $S = 300 \times 300$. Periodic boundary conditions were used as it has been shown that they reduce finite-size effects [25]. For each shape of triangle, $N = 100$ independent, saturated packings were generated. The mean packing fraction was equal to $\theta = (1/N)(\sum_{i=1}^{N} n_i/S)$, where n_i was the number of triangles in the ith packing (see Supplemental Material [26]). The estimated statistical error of θ, given by $\sigma(\theta) = (1/N)[\sum_{i=1}^{N}(\theta - n_i/S)^2]^{1/2}$, was below 10^{-4}. During packing generation, three helper segments from the mass center of the triangle to its vertices were used (see Fig. 5).

B. Packing fractions

Fragments of example packing are shown in Fig. 6. The analysis of the obtained values of the mean saturated packing fractions started with the case of isosceles triangles of side lengths 1, 1, x, where x can be any positive value. The results are shown in Fig. 7. The highest packing fraction is reached for $x = 0.63$ and equals $\theta = 0.552\,814 \pm 0.000\,063$. A slightly lower value is observed for $x = 1.31$, where $\theta = 0.550\,906 \pm 0.000\,057$. The local minimum is present for the equilateral triangle, which agrees with theoretical arguments that shapes of higher symmetry form less dense packings [20]. Similar behavior has been observed for other anisotropic shapes [12,14,16,17]. The packing fraction for the equilateral triangle is $\theta = 0.525\,818 \pm 0.000\,059$ which agrees with previously reported values [13,23].

The results for packings built of arbitrary triangles are shown in Fig. 8. Interestingly, for all studied sets of parameters (a, b) not a single packing was found that was denser than the one obtained for isosceles triangles. In addition, the maximum in the (a, b) plane seems to be almost flat along the direction that connects two maxima for isosceles triangles. Therefore, at this level of accuracy, it is hard to determine if
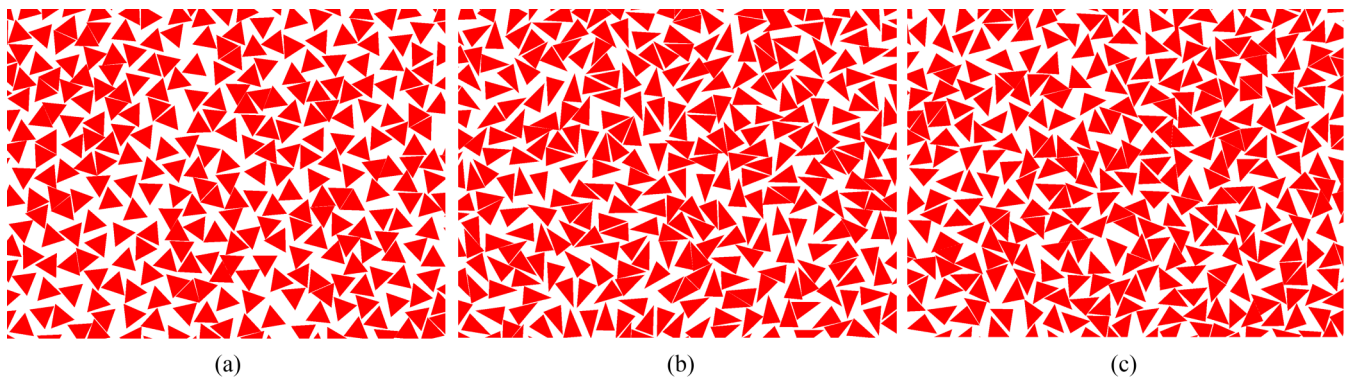

FIG. 6. Fragments of example saturated random packings of triangles for (a) $a = b = 1.0$; (b) $a = 0.63$, $b = 1.0$; and (c) $a = 1.0$, $b = 1.31$.
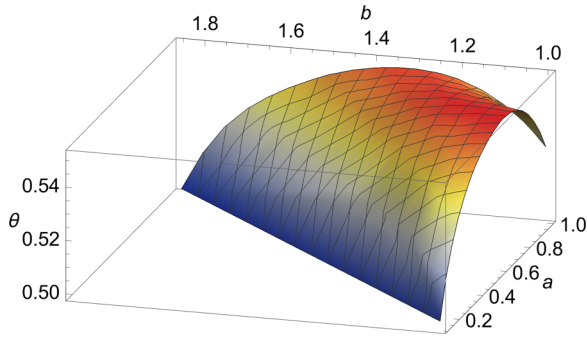
FIG. 8. The mean saturated packing fraction for arbitrary triangles.

the global maximum is reached for the isosceles triangle or for another shape that is close to ($a = 0.63$, $b = 1$). However, it is unexpected that the shape of higher symmetry does not correspond to a local minimum of the packing fraction, i.e., the packing fraction does not decrease towards edges given by $a = 1$ and $b = 1$ [20].

C. Microstructural properties

Microstructural properties of the obtained packings can be studied using density autocorrelation function:

$$G(r) = \lim_{dr \to 0} \frac{N(r, r + dr)}{2\pi r \, dr \, \theta}, \quad (4)$$

where $N(r, r + dr)$ is the mean number of triangles with the mass center at the distance between r and $r + dr$ from the center of a reference particle. The results for sample packings are shown in Fig. 9. The behavior observed is typical for random media. For the equilateral triangle the first maximum is higher and sharper due to the symmetry—when two triangles are touching their sides, the distance between their centers does not depend on which sides are in touch. For isosceles triangles, this is not true, and therefore instead of one maximum, two smaller maxima are observed. For larger distances, $G(r)$ oscillations are damped. It has been shown analytically that for one-dimensional RSA packings these oscillations vanish superexponentially with the distance [27]. Here, $G(r) \approx 1$ for $r > 5$.
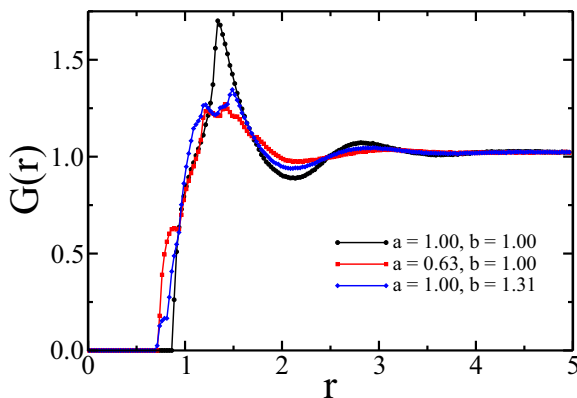

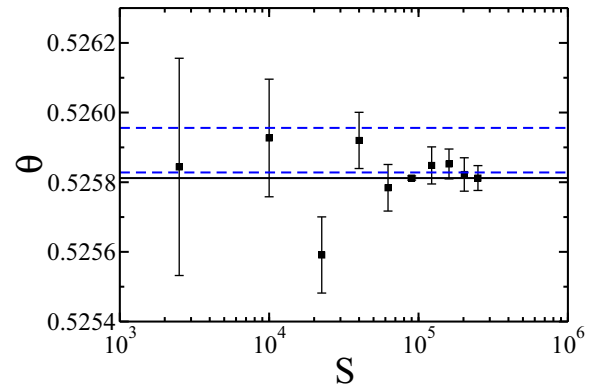FIG. 9. Density autocorrelation functions for packings built of three different triangles.

FIG. 10. The dependence of saturated mean packing fraction on packing size. For all packing sizes except $S = 90\,000$, packing fraction was calculated by averaging over 100 independent random packings. For $S = 90\,000$, 10^4 independent packings were used. Error bars correspond to the standard deviation of the mean packing fraction. Solid black line corresponds to $\theta(S = 9000) = 0.525\,811\,9$. Dashed blue lines correspond to the range of θ determined by Zhang [13].

D. Finite-size effects

It has been shown for the RSA of disks that the measured value of saturated packing fraction for small packings oscillates with respect to the system size, similar to the oscillations found in the tail of the autocorrelation function [25]. Therefore, it was expected (as a packing side length used in the study is much larger than 5) that the finite-size effects should not affect the obtained values of packing fraction. To prove that, the dependence of packing fraction on packing size for equilateral triangles was analyzed. The results are shown in Fig. 10. The result from our previously used system size, $S = 90\,000$, certainly agrees with the best estimation given by the solid black line. Here we also present the packing fraction range reported earlier in [13], which is slightly higher but very close to the one reported here. This can happen with approximately one-third probability due to the probabilistic interpretation of standard deviations.

E. Kinetics of packing growth

Kinetics of RSA packing growth near saturation depends on the number of iterations according to the power law [6,7]

$$\theta(t) = \theta - A t^{-1/d}. \quad (5)$$

The number of iterations is often measured in the so-called dimensionless time, which unit corresponds to S/S_p iterations, where S_p is the area covered by a single particle in the packing. Such a scaling does not affect the exponent in the above power law and is commonly used to compare RSA kinetics of differently sized packings. Although the parameter d in (5) is simply the space dimension for spheres, Baule has shown recently that when anisotropic particles are placed according to the RSA protocol in such a way that their centers lay on a one-dimensional line, d depends on properties of contact function of packed shapes. Parameter d is noticeably bigger when this function is not analytic [22]. For two-dimensional
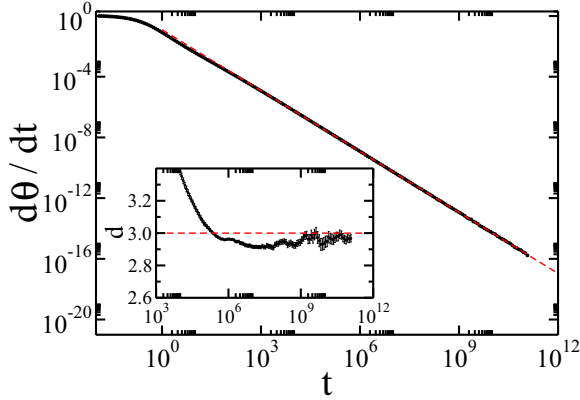
FIG. 11. The dependence of increments of packing fraction on dimensionless time for equilateral triangles. Black dots are measured data and red dashed line is a power-law fit $d\theta/dt = 0.127\,28\,t^{-1.3387}$. Inset shows the value of parameter d estimated from data obtained for times within $[0.01t, t]$. Red dashed line corresponds to $d = 3$.
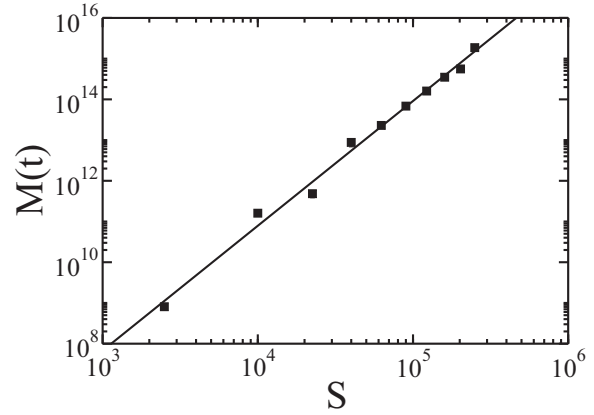

FIG. 12. The dependence of the median of the number of iterations (expressed in dimensionless time) needed to generate saturated RSA packing built of equilateral triangles on packing size. Squares represent numerical data obtained from 100 packings. Solid line is a power fit: $M(t) = 0.043\,177S^{3.0649}$.

nonspherical RSA packings, there is a numerical evidence that $d = 3$ for RSA of ellipses [8,17], or rectangles [28,29]; however, these studies do not reach saturation limit. In this study, using the algorithm described above, we can check the validity of (5) and calculate d arbitrarily close to saturation. For all examined shapes, the determined value of parameter d was in the range of $(2.79, 3.17)$ with the fitting error at the level of 0.05. Figure 11 shows example data obtained for saturated packings built of equilateral triangles. The slope of the dependence is almost constant (see inset in Fig. 11) for large enough t, and $d \approx 3$.

For saturated packing of spherically symmetric particles, it was previously established that parameter d can also be obtained from the dependence of the median of iterations needed to generate saturated packings expressed in dimensionless time units on packing size [24]:

$$M(t) \sim S^d. \tag{6}$$

As shown in Fig. 12, the above relation is also valid for equilateral triangles, and, because its derivation is based on very general assumptions [24], we expect it to be valid for other anisotropic shapes.

Figure 12 also confirms that d for triangles is the same as for other anisotropic shapes with analytic contact function. Thus, the result obtained by Baule for RSA on a one-dimensional line [22] cannot be directly extended to higher dimensions.

Lastly, we study the distributions of packing fractions and the number of iterations needed to generate saturated packings. They are important because the full distribution provides extra information besides the mean and standard deviation. Here, as we can see in Fig. 13, the packing fraction is normally distributed, while a power law gives the distribution of the number of iterations needed to saturate a packing. It is noteworthy that the same was observed for RSA of disks [24,30]. Thus again, this seems to be a universal property for RSA of arbitrary shapes.

IV. SUMMARY

This study presents an algorithm that allows generating saturated random packings according to the RSA protocol for a wide range of shapes. The algorithm is based on the idea recently introduced in [13], and incorporates an additional step of adding helper segments to a polygon in order to determine if another trial particle can be added to the packing in a given area.

This algorithm was used to generate saturated random packings of triangles to find the shape that yields the densest packings. It appeared that the highest packing fraction, $0.552\,814 \pm 0.000\,063$, was observed for isosceles triangles of side length ratios $0.63 : 1 : 1$. Also, a previous estimation of the mean packing fraction for equilateral triangles [13] is improved to $0.525\,811\,9 \pm 0.000\,005\,9$.
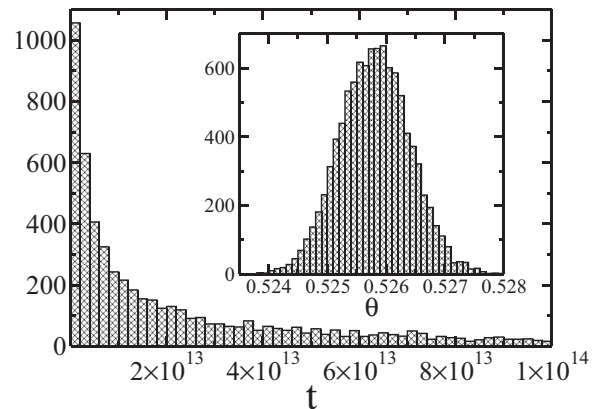

FIG. 13. Histograms of the number of iterations (expressed in dimensionless time) needed to generate saturated packing and the packing fraction (inset) obtained from studying 10^4 packings built of equilateral triangles.

The presented algorithm was also used to check kinetics of the growth of RSA packing built of triangles near saturation, and it was found that in contrast to one-dimensional RSA, it is governed by the same exponent as for other smooth shapes. Additionally, it has been shown that the number of iterations needed to saturate a packing scales with packing size according to a power law, which is the same as for spherically symmetric particles.

See Ref. [31] for the source code of the program generating saturated random packings of arbitrary polygons and, in particular, triangles.

[1] J. W. Evans, Rev. Mod. Phys. **65**, 1281 (1993).

[2] J. Talbot, G. Tarjus, P. Van Tassel, and P. Viot, Colloid. Surf. A **165**, 287 (2000).

[3] A. Renyi, Publ. Math. Inst. Hungar. Acad. Sci. **3**, 109 (1958).

[4] J. Feder, J. Theor. Biol. **87**, 237 (1980).

[5] S. Torquato, T. M. Truskett, and P. G. Debenedetti, Phys. Rev. Lett. **84**, 2064 (2000).

[6] Y. Pomeau, J. Phys. A **13**, L193 (1980).

[7] R. H. Swendsen, Phys. Rev. A **24**, 504 (1981).

[8] M. Cieśla, G. Pajaak, and R. M. Ziff, J. Chem. Phys. **145**, 044708 (2016).

[9] J.-S. Wang, Int. J. Mod. Phys. C **5**, 707 (1994).

[10] M. S. Ebeida, S. A. Mitchell, A. Patney, A. A. Davidson, and J. D. Owens, *Computer Graphics Forum*, Vol. 31 (Wiley Online Library, 2012), pp. 785–794.

[11] G. Zhang and S. Torquato, Phys. Rev. E **88**, 053312 (2013).

[12] K. Haiduk, P. Kubala, and M. Cieśla, Phys. Rev. E **98**, 063309 (2018).

[13] G. Zhang, Phys. Rev. E **97**, 043311 (2018).

[14] W. Kasperek, P. Kubala, and M. Cieśla, Phys. Rev. E **98**, 063310 (2018).

[15] M. Cieśla, J. Comput. Phys. **401**, 108999 (2020).

[16] R. D. Vigil and R. M. Ziff, J. Chem. Phys. **91**, 2599 (1989).

[17] P. Viot, G. Tarjus, S. Ricci, and J. Talbot, J. Chem. Phys. **97**, 5212 (1992).

[18] A. Donev, I. Cisse, D. Sachs, E. A. Variano, F. H. Stillinger, R. Connelly, S. Torquato, and P. M. Chaikin, Science **303**, 990 (2004).

[19] W. Man, A. Donev, F. H. Stillinger, M. T. Sullivan, W. B. Russel, D. Heeger, S. Inati, S. Torquato, and P. M. Chaikin, Phys. Rev. Lett. **94**, 198001 (2005).

[20] A. Baule, R. Mari, L. Bo, L. Portal, and H. A. Makse, Nat. Commun. **4**, 2194 (2013).

[21] P. Kubala, M. Cieśla, and R. M. Ziff, Phys. Rev. E **100**, 052903 (2019).

[22] A. Baule, Phys. Rev. Lett. **119**, 028003 (2017).

[23] M. Cieśla and J. Barbasz, Phys. Rev. E **90**, 022402 (2014).

[24] M. Cieśla, J. Stat. Phys. **166**, 39 (2017).

[25] M. Cieśla and R. M. Ziff, J. Stat. Mech. (2018) 043302.

[26] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.100.062901 for the *Mathematica* notebook which contains all measured mean packing fractions together with their standard deviations.

[27] B. Bonnier, D. Boyer, and P. Viot, J. Phys. A **27**, 3671 (1994).

[28] P. Viot and G. Tarjus, Europhys. Lett. **13**, 295 (1990).

[29] D. Vigil and R. M. Ziff, J. Chem. Phys. **93**, 8270 (1990).

[30] M. Cieśla and A. Nowak, Surf. Sci. **651**, 182 (2016).

[31] https://github.com/misiekc/polygonRSA.